IEEE/ACM
2023 INTERNATIONAL CONFERENCE ON COMPUTER-AIDED DESIGN
ICCAD
42nd Edition

iPL-3D: A Novel Bilevel Programming Model for Die-to-Die Placement

Xueyan Zhao[1,3,2] Shijian Chen[2,1] Yihang Qiu[4,2] Jiangkao Li[5,2]
Zhipeng Huang[2] Biwei Xie[1,2] Xingquan Li[5,2] Yungang Bao[1,3,2]

[1]Institute of Computing Technology, CAS [2]Peng Cheng Laboratory [3]University of Chinese Academy of Sciences
[4]Guangdong University of Technology [5]Minnan Normal University

Introduction

- 3D-ICs are expected to further improve chip performance after Moore's Law reaches its limits.
- The placement stage is more critical in 3D-IC flow than ever as it is a major contributor to further improvements in chip performance.
- Existing work is difficult to deal with the current complex process constraints and it is difficult to consider comprehensive objectives.

Problem Statement

The coordinate of the cell c_i is denoted by (x_i, y_i, z_i), where $z_i \in \{0, 1\}$. And $e_j \in E$ is crossing net if and only if it has both top and bottom cells. We note the bottom part as e_j^- and the top as e_j^+. In addition, $T = \{t_1, t_2, ..., t_m\}$ represents the set of terminals used by crossing nets. (x_{t_j}, y_{t_j}) denotes the coordinate of terminal t_j. We use $\mathrm{WL}_t(e_j, \cdot)$ and $\mathrm{WL}(e_j, \cdot)$ to represent the wirelength of net e_j in the 3D and 2D, respectively. Their relationship is defined as Eq. (1), and $\mathbf{x}_{e_j} = (\mathbf{x}, \mathbf{y}, x_{t_j}, y_{t_j})$.

$$\mathrm{WL}_t(e_j; \mathbf{x}, \mathbf{y}, \mathbf{z}, x_{t_j}, y_{t_j}) = \begin{cases} \mathrm{WL}(e_j^- \cup \{t_j\}; \mathbf{x}_{e_j}) + \mathrm{WL}(e_j^+ \cup \{t_j\}; \mathbf{x}_{e_j}) & \varepsilon(e_j; \mathbf{z}) = 1; \\ \mathrm{WL}(e_j; \mathbf{x}, \mathbf{y}) & \varepsilon(e_j; \mathbf{z}) = 0, \end{cases} \tag{1}$$

where $\varepsilon(e; \mathbf{z}) = \max_{c_i \in e}(z_i) - \min_{c_i \in e}(z_i)$. , and $\mathbb{I}(\cdot)$ is indicator function.

Therefore, the original D2D placement problem can be formalized as the optimization problem shown in Eq. (2)

$$\begin{aligned} \min_{\mathbf{x},\mathbf{y},\mathbf{z},\mathbf{x}_t,\mathbf{y}_t} \quad & \sum_{e_j \in E} WL_t(e_j; \mathbf{x}, \mathbf{y}, \mathbf{z}, x_{t_j}, y_{t_j}) + \rho\varepsilon(e; \mathbf{z}), \\ s.t. \quad & D_b(\mathbf{x}, \mathbf{y}, \mathbf{x}_t, \mathbf{y}_t, \mathbf{z}) \le M_b, \ \forall b \in S_b, \\ & \textstyle\sum_{i=1}^n A_1(c_i)\mathbb{I}(z_i) \le u_t A, \\ & \textstyle\sum_{i=1}^n A_0(c_i)\mathbb{I}(1 - z_i) \le u_b A, \\ & \textstyle\sum_{e_j \in E} \varepsilon(e_j; \mathbf{z}) \le N_t. \end{aligned} \tag{2}$$

Bilevel Programming

For the upper-level objective function $F : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}$ and lower-level objective function $f : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}$, the bilevel programming problem is given by

$$\begin{aligned} \min_{x_u \in X_U, x_l \in X_L} \quad & F(x_u, x_l) \\ s.t. \quad & x_l \in \arg\min_{x_l \in X_L} \{f(x_u, x_l) | \\ & \qquad g_j(x_u, x_l) \le 0, j = 1, ..., J\} \\ & G_k(x_u, x_l) \le 0, k = 1, ..., K, \end{aligned}$$

where $G_k : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}, k = 1, ..., K$ denote the upper-level constraints, and $g_j : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}$ represent the lower-level constraints, respectively. Equality constraints may also exist that have been avoided for brevity.

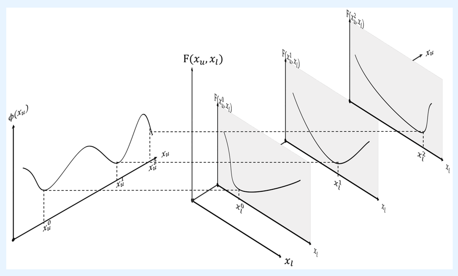
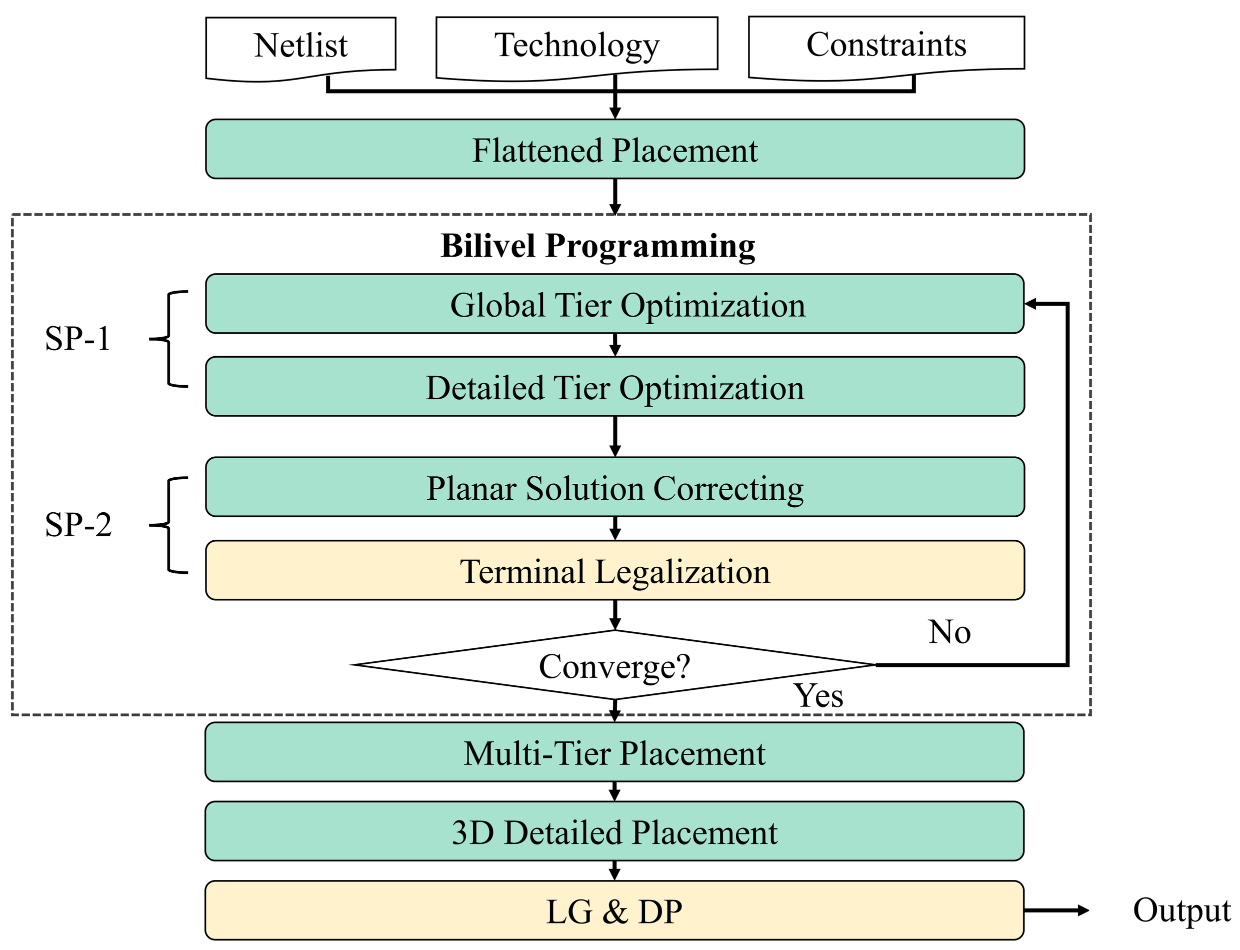


Figure 1. Graphical representation for optimal value function $\phi(\cdot)$ when the lower subproblem was solved.

Overall Flow - iPL-3D



Bilevel Programming Model for D2D Placement

Important Observation

- There is a natural dominance relationship among decision variables.
- Once z is determined, the remaining part is similar to the traditional 2D Placement problem.
- Traditional min-cut-based methods struggle to obtain a global view

Modeling

- The upper level variable corresponds to $\mathbf{z}$.
- The lower level variable corresponds to $\mathbf{x}_l = (\mathbf{x}, \mathbf{y}, \mathbf{x}_t, \mathbf{y}_t)$.
- The objective function can be rewrite as: $F(\mathbf{z}, \mathbf{x}_l) = WL(\cdot) + \rho\varepsilon(\cdot)$
- The lower level problem can be defined as:

$$g(\mathbf{z}) = \min_{\mathbf{x}_l} \{F(\mathbf{x}_l, \mathbf{z}) | \ D_b(\mathbf{x}_l, \mathbf{z}) \le M_b, \forall b \in S_b\}$$

$$\Psi(\mathbf{z}) = \arg\min_{\mathbf{x}_l} \{F(\mathbf{x}_l, \mathbf{z}) | \ D_b(\mathbf{x}_l, \mathbf{z}) \le M_b, \forall b \in S_b\}$$

- There is a tautology: $\forall \mathbf{x}_l^* \in \Psi(\mathbf{z}), g(\mathbf{z}) = F(\mathbf{x}_l^*, \mathbf{z})$. Then we use $g(\mathbf{z})$ replace the original objective function $F(\mathbf{x}_l^*, \mathbf{z})$. The original problem can be rewritten as Eq. 3.

$$\begin{aligned} \min_{\mathbf{z}, \mathbf{x}_l} \quad & g(\mathbf{z}) \\ s.t. \quad & \mathbf{x}_l \in \Psi(\mathbf{z}) \\ & \textstyle\sum_{i=1}^n A_1(c_i)\mathbb{I}(z_i) \le u_t A \\ & \textstyle\sum_{i=1}^n A_0(c_i)\mathbb{I}(1 - z_i) \le u_b A \\ & \textstyle\sum_{e_j \in E} \varepsilon(e_j; \mathbf{z}) \le N_t \end{aligned} \tag{3}$$

Solve Tow Subproblems Alternately

- The variable $\mathbf{x}_l$ does not appear in other constraints and objectives.
- To solve efficiently, we split the original problem and introduce a surrogate function.

$$\begin{aligned} \min_{\mathbf{z}} \quad & \hat{g}(\mathbf{x}_l^k, \mathbf{z}) \\ s.t. \quad & \textstyle\sum_{i=1}^n A_1(c_i)\mathbb{I}(z_i) \le u_t A \\ & \textstyle\sum_{i=1}^n A_0(c_i)\mathbb{I}(1 - z_i) \le u_b A \\ & \textstyle\sum_{e_j \in E} \varepsilon(e_j; \mathbf{z}) \le N_t \end{aligned} \tag{4}$$

$$\mathbf{x}_l^{k+1} = \underset{\Psi(\mathbf{z}^{k+1})}{Proj} (\mathbf{x}_l^k) \tag{5}$$

Flattend Placement

- **Motivation**: A high-quality solution can also provide sufficient information for the surrogate function $\hat{g}(\mathbf{x}_l, \mathbf{z})$
- **Method**: Place all standard cells in one layer and double the capacity of the bin. Then solve the global placement problem to obtain $\mathbf{x}_{2D}$
- **Theorem**: The quality of the optimal planar solution obtained from Flattened Placement is the upper bound for the final 3D solution.

$$WL(x_{2D}^*) \le WL(x_{2 \to 3D}) \le WL(x_{3D}^*)$$

Tier Optimization

Motivation

- **Consider MIV Density & Wirelength Partitioning**: Changes in the vertical coordinates not only affect the number of terminals but also lead to additional wirelength changes caused by terminals.
- **Optimized from two perspectives of coarse-grained and fine-grained**: Coarse-grained can provide a relatively good initial solution, while fine-grained can further refinement.

Global Tier Optimization

- **Best Improvement Algorithm**: The gain is maintained using a priority queue, and the candidate cell with the largest gain is iteratively selected for tier changing.
- **Parameterized Comprehensive Surrogate Function**: When γ is sufficiently large, select the region with the highest density, and sort the remaining parts within the region based on their weights.

$$\begin{aligned} p(S \cup \{c_i\}) - p(S) = & \ \Delta\text{wirelenth} + \rho\Delta\#\text{Terminal} \\ & + \alpha\big(d(S \cup \{c_i\}) - d(S)\big) \\ & + \beta\big(o(S \cup \{c_i\}) - o(S)\big) \\ & - \gamma d(S), \end{aligned} \tag{6a}$$

where

$$d(S) = \sum_{\text{region } r} \max(\frac{A_r - M_r}{h_r}, 0),$$
$$o(S) = \sum_{i=1}^{n} \sum_{c_j \in \{c_j | \forall c_j \in V, z_j = z_i\}} \frac{\mathrm{Overlap}(c_j, c_i)}{h_{c_i}} \tag{6b}$$

- Knapsack maximization like priority calculation:

Tier Optimization

Detailed Tier Optimization

- **Dynamic Row-based Data Structure**: Implement the insertion and deletion of units at any position in a row.
- **Simple 3D Detailed Placement**: Quickly generate legal solutions and calculate actual gains.

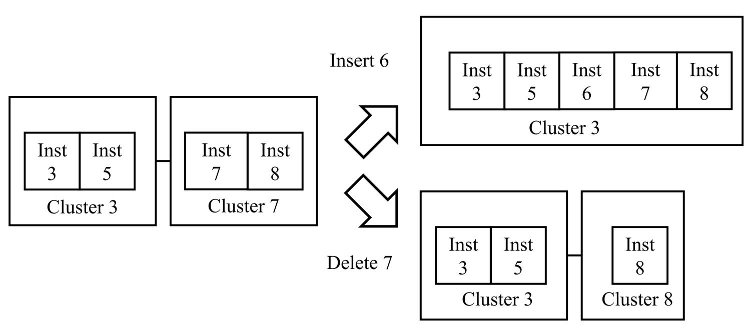


Figure 2. Example for Row-based Data-structure.

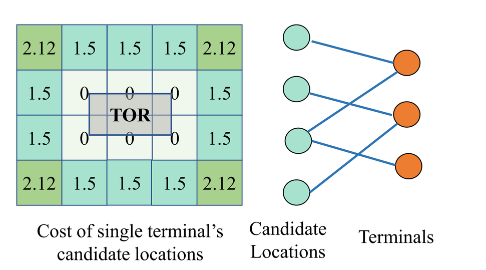


Figure 3. Example for Terminal Legalization.

Terminal Legalization

Problem Characteristics

- Terminals are of the same size.
- Cost calculation is independent.

Solving

- **Bipartite Graph Matching**: Select k candidate positions around each terminal in its optimal region. Then, network simplex algorithm is performed to do bipartite graph matching between terminal and candidate locations .
- **Lower Bound**: $WL(x_{real}^*) \le WL(x_{grid}^*) + 2C\#\text{terminal}$.

Experimental Results

- Compared to the top three competitors, there is an improvement in wirelength of 4.33%, 4.42%, and 5.88%, respectively. The speed is 1.84x faster than the first-place competitor.
- #Terminals used is the lowest, with improvements of 79.61%, 16.74%, and 15.76% compared to the top three competitors.
- The final result shows an increase in wirelength of 7.63% compared to Flatten GP (Theorem 1).

Table 1. Experimental Results on ICCAD 2022 Contest Benchmarks.

Case	Flattened	3th			2nd			1st			Ours		
	GP	HPWL	#Terminal	CPU(s)	HPWL	#Terminal	CPU(s)	HPWL	#Terminal	CPU(s)	HPWL	#Terminal	CPU(s)†
case2	1758214	2097487	**163**	10	2080647	477	14	2072075	1131	45	**1992499**	461	45
case2_h	2111322	2644791	**151**	9	2735158	687	15	2555461	1083	40	**2530195**	658	53
case3	26474613	33063568	14788	145	30969011	11257	437	30580336	16820	635	**30234112**	**9612**	442
case3_h	24200040	28372567	11211	133	27756492	8953	482	27650329	16414	412	**26939286**	**8203**	479
case4	248129463	281378079	46468	925	274026687	51480	3284	281315669	84069	2580	**267381744**	**43140**	1078
case4_h	272085522	307399565	58860	983	308359159	59896	3283	301193374	84728	2239	**289541474**	**51641**	1144
N.Total	-7.63%	5.88%	15.76%	0.68	4.42%	16.74%	2.32	4.33%	79.61%	1.84	**0.00%**	**0.00%**	1.00

Table 2. Results With and Without Alternating Optimization

Case	w/o. Alternating Optimization.			w/ Alternating Optimization		
	HPWL	#Terminal	CPU(s)	HPWL	#Terminal	CPU(s)
case2	2032655	555	20	1992499	461	45
case2_h	2562890	793	19	2530195	658	53
case3	30332531	10604	135	30234112	9612	442
case3_h	26935732	9288	128	26939286	8203	479
case4	270042122	54112	604	267381744	43140	1,078
case4_h	294923683	63283	637	289541474	51641	1,144
N.Total	1.33%	21.91%	0.48	0.00%	0.00%	1.00

Table 3. Terminal Legalization Experimental Results.

Case	C	#Terminal	WL	CPU(s)	TOR	Ratio
case2	200	461	1992499	1	1981785	0.54%
case2_h	228	658	2530195	1	2512837	0.69%
case3	100	9612	30234112	7	30141038	0.31%
case3_h	92	8203	26939286	5	26875050	0.24%
case4	124	43140	267381744	15	266850007	0.20%
case4_h	132	51641	289541474	16	288659033	0.30%

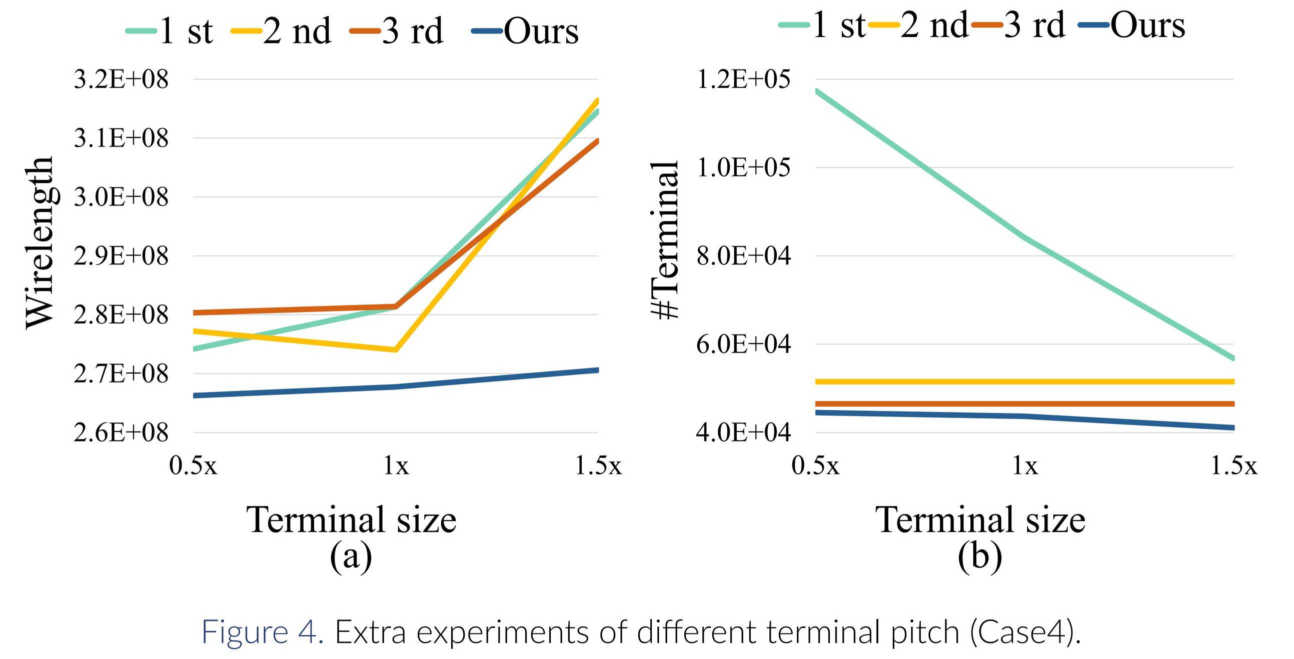


Figure 4. Extra experiments of different terminal pitch (Case4).

References

[1] Kai-Shun Hu, I-Jye Lin, Yu-Hui Huang, Hao-Yu Chi, Yi-Hsuan Wu, and Chin-Fang Cindy Shen. 2022 ICCAD CAD contest problem B: 3D placement with D2D vertical connections. In *Proceedings of ICCAD*, pages 1–3. IEEE, 2022.